

<LEI\_ESII2021\_TP2\_GRUP05>  
TEST DESIGN SPECIFICATION  
CLASS PAYMENTTRANSACTION

---

Version <1.2>  
<11/01/2022>

### Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Rui Neto	06-01-2022	Simão Santos	06-01-2022	Início da doc.
1.1	Simão Santos	07-01-2022	Rui Neto	07-01-2022	Fim da doc.
1.2	Rui Neto	11-01-2022	Simão Santos	11-01-2022	Alteração da doc.

# Índice

1. Introdução .....	4
1.1. Identificador do documento .....	4
1.2. Âmbito .....	4
1.3. Referências .....	4
2. Features/Itens a testar .....	4
3. Detalhes da abordagem aos testes .....	5
Método Construtor PaymentTransaction() .....	5
Método addTransactionLine() .....	5
Método getOrderId() .....	5
4. Identificação dos Testes .....	6
5. Critérios de passagem ou falha das features .....	6

## 1. Introdução

### 1.1. Identificador do documento

[TCS-9]PaymentTransaction

### 1.2. Âmbito

O documento foca-se na testagem das funcionalidades da Class PaymentTransaction.

Items no âmbito: Testes funcionais da class PaymentTransaction.

Items fora do âmbito: Testes à Base de dados, hardware, alguma interface externa e performance.

### 1.3. Referências

Sem documentos a referenciar.

## 2. Features/Itens a testar

Item a testar	Descrição	UserStory	Responsabilidade
PaymentTransaction()	Método construtor. Recebe como parâmetros duas Market Place Entitys.	#1	Alta
addTransactionLine()	Permite adicionar uma e apenas uma TransactionLine à PaymentTransaction.	#1	Alta
getOrderId()	Devolve o ID da order onde está incluída a transação.	#1	Alta

### 3. Detalhes da abordagem aos testes

#### Método Construtor PaymentTransaction()

##### Pré-requisitos

- Existem duas instâncias válidas da classe MarketPlaceEntity.

##### Classes de equivalência

EC1 -> Lança exceção caso a primeira entity passada como parâmetro seja nula.

EC2 -> Não lança exceção caso a primeira entity passada como parâmetro seja nula.

EC3 -> Lança exceção caso a segunda entity passada como parâmetro seja nula.

EC4 -> Não lança exceção caso a segunda entity passada como parâmetro seja nula.

Condição	Classe de Eq. válida	Classe de Eq. inválida
Valor Inputs	EC1, EC3	EC2, EC4

#### Método addTransactionLine()

##### Pré-requisitos

- O input é do tipo TransactionLine

##### Classes de equivalência

EC1 -> Retorna falso e o counter é maior que 0

EC2 -> Não retorna falso e o counter é maior que 0

Condição	Classe de Eq. válida	Classe de Eq. inválida
Valor retornado	EC1	EC2

#### Método getOrderld()

##### Pré-requisitos

- Existe uma instância da classe PaymentTransaction

##### Classes de equivalência

EC1 -> O output é do tipo 'String'

EC2 -> O output não é do tipo 'String'

EC3 -> Retorna o 'orderld' corretamente

EC4 -> Não retorna o 'orderld' corretamente

Condição	Classe de Eq. válida	Classe de Eq. inválida
Tipo do output	EC1	EC2
Valor retornado	EC3	EC4

## 4. Identificação dos Testes

TestCase	Método	Descrição
TC_PAYT_1	PaymentTransaction()	Verificar resultados ao instanciar uma PaymentTransaction com o primeiro parâmetro nulo.
TC_PAYT_2	PaymentTransaction()	Verificar resultados ao instanciar uma PaymentTransaction com o segundo parâmetro nulo.
TC_PAYT_3	addTransactionLine()	Verificar resultado ao adicionar mais que uma linha de transação à transação.
TC_PAYT_4	getOrderId()	Verificar se retorna corretamente o ID da order.

## 5. Critérios de passagem ou falha das features

Features	Critérios de passagem
Criar PaymentTransaction	É possível criar uma PaymentTransaction. Impossibilita a criação de uma PaymentTransaction com a primeira entity nula. Impossibilita a criação de uma PaymentTransaction com a segunda entity nula.
Adicionar linhas de transação	Permite adicionar uma e apenas uma linha de transação.
Retornar informações da PaymentTransaction	Retorna corretamente o ID da Order.